ИСПОЛЬЗОВАНИЕ ГРАФИЧЕСКОГО УСКОРИТЕЛЯ ARM ДЛЯ ОПТИМИЗАЦИИ ВЫВОДА НЕЙРОННЫХ СЕТЕЙ В INTEL OPENVINO TOOLKIT[1*]

А.Ю. Нестеров, И.Б. Мееров

Нижегородский государственный университет им. Н. И. Лобачевского

Область глубоких сверточных нейронных сетей активно развивается, что приводит к потребности быстрой работы вывода (прямой проход алгоритма обратного распространения ошибки) нейронных сетей на различных устройствах. Одна из востребованных областей применения нейронных сетей - мобильные технологии. Мобильные устройства часто являются маломощными, что требует рационального использования памяти и процессорных ресурсов. Для того, чтобы вывод нейронных сетей работал в режиме реального времени, чаще всего используют графический ускоритель в ARM-архитектурах. В данной работе будут описаны идеи работы по интеграции вывода нейронных сетей в Intel OpenVINO Toolkit на графических ускорителях ARM.

Ключевые слова: advanced risk machine, opencl, графический ускоритель, гетерогенное программирование.

1. Введение

Область искусственного интеллекта (ИИ) с каждым днем становится все более востребованной. Методы ИИ используются на практике при решении широкого спектра задач от проверки качества производства на заводах до анализа данных в программном обеспечении медицинского оборудования. Программная реализация методов ИИ, ориентированная на современную вычислительную технику, требует не только глубокого понимания теоретических основ данной предметной области, но и разработки соответствующих подходов, технологий, и инструментария.

В рамках данной работы этот вопрос изучается в контексте использования мобильных устройств (планшетов, смартфонов и другой аналогичной техники). В сравнении с мощным серверным оборудованием мобильные устройства обладают ограниченными ресурсами, однако спектр задач, в которых они могут и должны с успехом применяться, очень широк. К числу востребованных приложений методов и программных средств ИИ на мобильных устройствах относится, например, телемедицина. Потребители подобных приложений могут с использованием методов ИИ провести первичную диагностику и, при необходимости, вовремя связаться с врачом для дальнейшего обследования.

Вопросы экономии ресурсов времени и памяти при работе методов ИИ на мобильных устройствах являются достаточно важными и нередко определяют саму возможность использования таких программно-аппаратных конфигураций для решения задач. Одним из программных пакетов, решающих обширное число задач ИИ, является открытая разработка компании Intel - OpenVINO Toolkit [1]. Эта разработка рассматривается нами в качестве базовой. Главным ее недостатком в контексте целей работы является отсутствие поддержки графических процессоров, позволяющих значительно ускорить вычисления в современных смартфонах. Наше исследование направлено на то, чтобы ускорить обработку задач ИИ, используя графический ускоритель смартфона. Для этого требуется разработать специальный программный модуль.

[1*] Исследование поддержано НЦМУ «Центр фотоники» (соглашение № 075-15-2020-927).

2. Краткий обзор Intel OpenVINO Toolkit

2.1 Этапы работы с выводом в Intel OpenVINO Toolkit

Работа с выводом в Intel OpenVINO Toolkit разбивается на несколько основных стадий:

1. **Конвертация претренированной модели** нейронной сети в промежуточное представление (Intermediate Representation) с помощью кроссплатформенного инструмента Model Optimizer [2]. Model Optimizer не только преобразует модель в промежуточное представление, но и выполняет ряд оптимизаций. К примеру, ряд операций могут объединяться или переопределяться для упрощения графа сети.
2. **Вывод промежуточного представления** в C++ библиотеке Inference Engine [3]. Inference Engine – это набор библиотек C++, предоставляющих общий API для создания решений вывода для любой представленной платформеы. Для ARM=архитектур сейчас поддерживается только ARM CPU [4].
3. **Оптимизация производительности вывода** нейронной сети с помощью POT (Post-training Optimization Tool). POT предназначен для оптимизации вывода моделей глубокого обучения путем применения специальных методов без переобучения или тонкой настройки модели.

2.2 Способы интеграции вывода для собственной архитектуры вычислительных систем в библиотеке Inference Engine

Для интеграции нужного нам вывода на требуемом устройстве требуется знать, какие есть способы интеграции в библиотеке Inference Engine.

Существуют следующие способы интеграции:

1. *Механизм расширения вывода*. Данный подход уместен тогда, когда часть операций сверточной нейронной сети уже реализована на стороне библиотеки Inference Engine, а остальные части интегрируются с помощью данного механизма.
2. *Создание нового плагинного модуля для собственной вычислительной архитектуры*. Данный подход является оптимальным для случая, когда нет реализации вывода в библиотеке Inference Engine.

Для нашей работы подходит второй вариант интеграции вывода. В связи с отсутствием поддержки графического ускорителя ARM данный подход более подробно мы и рассмотрим далее. Назовем будущий плагин ARM GPU.

Архитектура плагинов библиотеки Inference Engine позволяет разрабатывать и подключать независимые решения, предназначенные для различных устройств. Плагин представлен в виде динамической библиотеки, использующей функцию **CreatePluginEngine**, которая позволяет создать новый экземпляр плагина.

Библиотека Inference Engine состоит из нескольких основных классов, которые надо реализовать, чтобы получить полнофункциональный вывод:

1. Класс Plugin:
   a. Предоставляет информацию об устройстве, которое плагин поддерживает.
   b. Класс создает исполняемый экземпляр сети, который представляет собой графовую структуру, специфичную для бэкенда нейронной сети для конкретного устройства (бэкенд ARM GPU плагина будет описан позднее).
2. Класс Executable Network:
   a. Представляет собой реализацию исполнения вывода, учитывая специфику устройства.
   b. Может создавать несколько экземпляров Inference Request.
3. Класс Inference Request:
   a. Запускает последовательно конвейер вывода.
   b. В данном классе можно делать замеры производительности вывода.
4. Класс Asynchronous Inference Request :
   a. Является оберткой класса Inference Request и параллельно запускает этапы конвейера вывода.
   b. Можно использовать несколько устройств в связке.

3. Бэкенд графического ускорителя ARM

Существование Intel OpenVINO ARM CPU плагина и возможность потенциальных гетерогенных вычислений центрального процессора ARM и графического ускорителя ARM определило выбор бэкенда ARM GPU плагина для библиотеки Inference Engine – это библиотека ARM Compute Library [5]. Данная библиотека предоставляет оптимизированные примитивы операций для вывода нейронных сетей на векторных инструкциях NEON и SVE (Scalable Vector Extension) для центрального процессора и оптимизированные ядра, написанные на стандарте OpenCL для графического ускорителя ARM. Последний факт определил будущий ARM GPU плагин в Inference Engine как OpenCL-подобный плагин [6].

Заключение

В данной работе были описаны идеи создания ARM GPU плагина в Intel OpenVINO Toolkit библиотеке Inference Engine для вывода нейронных сетей на графических ускорителях ARM-архитектур, а также общие подходы работы с Intel OpenVINO Toolkit.

На данный момент времени происходит создание архитектуры ARM GPU плагина. Далее планируется перейти к реализации плагина в библиотеке Inference Engine с использованием библиотеки ARM Compute Library в качестве бэкенда.

Литература

1. Репозиторий Intel OpenVINO Toolkit. URL: https://github.com/openvinotoolkit/openvino.
2. Документация Model Optimizer. URL: https://docs.openvino.ai/latest/openvino_docs_MO_DG_Deep_Learning_Model_Optimizer_DevGuide.html .
3. Документация Inference Engine: URL: https://docs.openvino.ai/latest/openvino_docs_IE_DG_Deep_Learning_Inference_Engine_DevGuide.html.
4. Github репозиторий Intel OpenVINO ARM CPU плагина. URL: https://github.com/openvinotoolkit/openvino_contrib/tree/master/modules/arm_plugin.
5. Sun D., Liu S., Gaudiot J. L. Enabling embedded inference engine with ARM compute li-brary: A case study //arXiv preprint arXiv:1704.03751. – 2017.
6. Документация о типах плагинов в Inference Engine. URL: https://docs.openvino.ai/latest/openvino_docs_ie_plugin_dg_plugin.html.